ISSN:0975-9646

Shubhashree Savant / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5898-5900

A Review on Edge Detection Techniques for Image Segmentation

Shubhashree Savant

Assistant Professor, MCA Department, MIT(E),
Dr. Babasaheb Ambedkar Marathwada University, Aurangabad
Maharashtra, India

Abstract— **Image segmentation is an important step of the digital image processing. Segmentation is a process of subdividing an image into the constituent parts or objects in the image. Segmentation is either discontinuity based or region based. Edge detection is useful for discontinuity based image segmentation technique. This paper presents a review on different edge detection techniques for image segmentation.**

Keywords— **Image segmentation, Edge detection, Gradient, Laplacian, Canny**

I. INTRODUCTION

The edges of image have characterized the boundaries and regions of the image. Edges in images are areas with strong intensity contrasts – a jump in intensity from one pixel to the next. Edge detection is one of the most important tasks in image processing and image analysis.[1] Edge detection process significantly reduces the amount of data and filter out the useless information while preserving important structural properties in an image. According to review work there are many ways to perform edge detection.

In general the edge detection techniques are grouped into two categories:

- Gradient
- Laplacian

II. LITERATURE SURVEY

The edge detection process serves to simplify the analysis of images by drastically reducing the amount of data to be processed while at the same time preserving useful structural information about object boundaries.[2] The filters are used in the process of identifying the image by locating the sharp edges which are discontinuous. These discontinuities bring changes in pixels intensities which define the boundaries of the object.[3] The edge detection aims to identify points in a digital image at which the image brightness changes sharply or abruptly. Image edge detection mainly deals with extracting edges in an image by identifying pixels where the intensity variation is very high.[4] Edges are used to measure the size of objects in an image; to isolate particular objects from their background; to recognize or classify objects.[5]

III. IMAGE SEGMENTATION

Segmentation subdivides an image into its constituent regions or objects. The level to which the subdivision is carried depends on the problem being solved. That is, segmentation should stop when the objects of interest in an application have been isolated.

Segmentation of nontrivial images is one of the most difficult tasks in image processing. Segmentation accuracy determines the eventual success or failure of computerized analysis procedures.

Image segmentation algorithms generally are based on one of two basic properties of intensity values:

1. Discontinuity
2. Similarity

In the first category, the approach is to partition an image based on abrupt changes in intensity, such as edges in an image.

The principal approaches in the second category are based on partitioning an image into regions that are similar according to a set of predefined criteria.

Detection of Discontinuities

There are three basic types of discontinuities in an digital image:

- Points
- Lines
- Edges.

IV. EDGE DETECTION

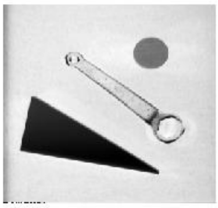
Edge detection is the most common approach for detecting meaningful discontinuities in gray level. Different approaches are given for implementing first-and second-order digital derivatives for the detection of edges in an image. Various types of edges are shown in Fig.1

Fig1. Types of edges (a) Step edge (b) Ramp edge (c) Line edge (d) Roof edge

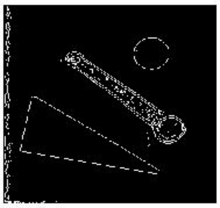
Edges are represented in images as sudden disparities. An ideal edge would be a unit step from one pixel to the next. However, many factors must be taken into consideration:

- Noise
- Brightness of the image as a whole
- Corners
- Edge-like features

A problem of fundamental importance in image analysis is edge detection. Edges characterize object boundaries and are therefore useful for segmentation, registration, and identification of objects. A typical application where edge detection techniques are involved is automatic character recognition.



Original Image



Edge Detection

The three fundamental steps of edge detection[6]

- **Image Smoothing:** suppress as much noise as possible, without destroying the true edges.
- **Detection of edge points:** determine which edge pixels should be discarded as noise and which should be retained (usually, thresholding provides the criterion used for detection).
- **Edge localization:** determine the exact location of an edge (sub-pixel resolution might be required for some applications, that is, estimate the location of an edge to better than the spacing between pixels). Edge thinning and linking are usually required in this step.

V. EDGE DETECTION TECHNIQUES

Edge detection techniques are grouped into two categories:

- Gradient
- Laplacian

First-order derivatives in an image are computed using the gradient. Second-order derivatives are obtained using the Laplacian

Edge Detection using function edge

The basic idea behind edge detection is to find places in an image where the intensity changes rapidly, using one of two general criteria:[7]

1. Find places where the first derivative of the intensity is greater in magnitude than a specified threshold.
2. Find places where the second derivative of the intensity has a zero crossing.

The general syntax for edge function is:

[g, t] = edge(f, 'method' ,parameters)

where f is the input image, method is one of edge detection technique and parameters are additional parameters.

Gradient

Edges in an image are pixel locations with abrupt changes in gray levels. In a continuous image (i.e., assuming continuous values), the derivative of the image f(x,y) assumes a local maximum in the direction of the edge. Therefore, one edge detection technique is to measure the gradient of f in a particular location. This is accomplished by using a gradient operator. Such operators, also called masks, provide finite-difference approximations of the orthogonal gradient vector fx and fy.

Sobel Edge Detector

The Sobel edge detector uses the masks shown in Fig. 2 to approximate digitally the first derivatives Gx and Gy and finds edges using the Sobel approximation to the derivatives.

Gx

-1	-2	-1
0	0	0
1	2	1

Gy

-1	0	1
-2	0	2
-1	0	1

Fig. 2

The general calling syntax for the Sobel detector is

[g, t] = edge(f, 'sobel', T, dir)

where f is the input image, T is a specified threshold, and dir specifies the preferred direction of the edges detected. 'horizontal', 'vertical' or 'both' (the default)

Robert Edge Detector

The Roberts edge detector uses the masks in Fig. 3 to approximate digitally the first derivatives Gx and Gy.

Gx

-1	0
0	1

Gy

0	-1
1	0

Fig. 3

Its general calling syntax is

[g, t] = edge(f, 'Roberts', T, dir)

The parameters of this function are identical to the Sobel parameters. The Roberts detector is one of the oldest edge detectors in digital image processing

Prewitt Edge Detector

The Prewitt edge detector uses the masks in Fig. 4 to approximate digitally the first derivatives Gx and Gy.

Gx

-1	-1	-1
0	0	0
1	1	1

Gy

-1	0	1
-1	0	1
-1	0	1

Fig. 4

Its general calling syntax is:

[g, t] = edge(f, 'Prewitt', T, dir)

The parameters of this function are identical to the Sobel parameters. The Prewitt detector is slightly simpler to implement computationally than the Sobel detector, but it tends to produce somewhat noisier results

Laplacian

The gradients work best when the gray level transition is quite abrupt. For smoother transitions, it is more advantageous to also compute the second-order derivative and to see when this second derivative crosses zero. If the second derivative crosses zero, it means that the location does indeed correspond to a maximum and therefore this pixel location is an edge location. This technique is called the zero-crossing edge location technique. A common mask operator for the estimation of the second derivative is the Laplacian operator

Laplacian of Gaussian (LoG)

The Laplacian is a 2-D isotropic measure of the 2^{nd} spatial derivative of an image. The image highlights regions of rapid intensity change and is therefore often used for edge detection. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian Smoothing filter in order to reduce its sensitivity to noise. The operator normally takes a single gray level image as input and produces another gray level image as output. [7] The general calling syntax for the LoG detector is

[g, t] = edge(f, 'log', T, sigma)

where sigma is the standard deviation.

Zero Crossing

It uses second derivative and it includes Laplacian operator. This is sensitive to noise. Haralick proposed the use of zero crossing of the second directional derivative of the image intensity function. The general calling syntax is:

[g, t] = edge(f, 'zerocross', T, H)

where H is the filter function.

Canny Edge Detector

Canny operator is based on three criteria.[2]The basic idea uses a Gaussian function to smooth image firstly. Then the maximum value of first derivative also corresponds to the minimum of the first derivative. In other words, both points with dramatic change of gray-scale (strong edge) and points with slight change of gray-scale (weak edges) corresponds to the second derivative zero crossing point. Thus these two thresholds are used to detect strong edges and weak edges.

VI. Conclusions

This paper presents a review on various edge detection techniques like gradient-based and laplacian-based techniques. Edge detection is a significant task for image segmentation used for object detection and many other applications. First-order derivatives in an image are computed using the gradient. Second-order derivatives are obtained using the Laplacian Result of edge detection techniques vary from set of images.

References

[1] A. Koschan and M. Abidi. *"Detection and Classification of Edges in Color Images"*. IEEE Signal Processing, 22(1):64–73, January 2005

[2] J. Canny, *"A Computational Approach to Edge Detection,"* IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 8, pp. 679-698, 1986

[3] G.T. Shrivakshan, Dr.C. Chandrasekar, *"A Comparison of various Edge Detection Techniques used in Image Processing"* IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 1, September 2012

[4] Er. Komal Sharma, Er. Navneet Kaur *"Comparative Analysis of Various Edge Detection Techniques"* IJARCSSE Volume 3, Issue 12, December 2013

[5] R.Jayakumar, B.Suresh "A REVIEW ON EDGE DETECTION METHODS AND TECHNIQUES" IJARCCE vol. 3, Issue 4, April 2014

[6] Gonzalez, R.C., and Woods, R. E. *"Digital Image Processing"* . 3rd ed., Pearson Education (Singapore) [1992]

[7] 34.Maar, D., Hildreth E., *"Theory of edge detection"* , Proceedings Royal Soc. London, vol. 207, 187-217, 1980

[8] S.Lakshmi, Dr. V.Sankaranarayanan "A study of Edge Detection Techniques for Segmentation Computing Approaches" IJCA Special issue on CASCT, 2010